The following bits represent:

WRITING THE REGISTERS
To write the FE registers, you need to do an array write to the CCM, the CCM takes the information and does the write of values to the FE board.

A: w - write array, you could have a put(single word write), get(single word write).
As a note you use array writes(w command) to communicate with FE boards I2C registers.
You have to use array write even if writing only one FE register.
B: Register 0 of the CCM, must be 0 for an array write, for a put or get use the register number.
C: Core number - use the excel file: CCM_I2C device_MEMORY_MAP.xls that details the systems core numbers,
D: Number of writes that you are doing
E: I2C address(basically selects the board/IC combination that you want to communicate with.
F: Beginnig register number that you want to write to.
G-J: the value you are writing into the registers.

A B C D E F G H I J

```
w 0 1 4 0  0 32 0 0 119; * write  datas to i2c core1 board1  CCA1 registers 0 - 3
w 0 1 4 1  0 32 0 0 119; * write  datas to i2c core1 board1  CCA2 registers 0 - 3
w 0 1 4 2  0 32 0 0 119; * write  datas to i2c core1 board1  CCA3 registers 0 - 3
w 0 1 4 16 0 32 0 0 119; * write  datas to i2c core1 board2  CCA1 registers 0 - 3
w 0 1 4 17 0 32 0 0 119; * write  datas to i2c core1 board2  CCA2 registers 0 - 3
w 0 1 4 18 0 32 0 0 119; * write  datas to i2c core1 board2  CCA3 registers 0 - 3
w 0 1 4 32 0 32 0 0 119; * write  datas to i2c core1 board3  CCA1 registers 0 - 3
w 0 1 4 33 0 32 0 0 119; * write  datas to i2c core1 board3  CCA2 registers 0 - 3
w 0 1 4 34 0 32 0 0 119; * write  datas to i2c core1 board3  CCA3 registers 0 - 3
w 0 2 4 0  0 32 0 0 119; * write  datas to i2c core2 board1  CCA1 registers 0 - 3
...
w 0 4 4 1  0 32 0 0 119; * write  datas to i2c core3 board1  CCA2 registers 0 - 3
...
w 0 8 4 17 0 32 0 0 119; * write  datas to i2c core4 board2  CCA2 registers 0 - 3
...
w 0 16 4 0 0 32 0 0 119; * write  datas to i2c core5 board1  CCA1 registers 0 - 3
...
w 0 32 4 33 0 32 0 0 119; * write  datas to i2c core6 board3  CCA2 registers 0 - 3
```

Note:
1.You can write more or less than 4 registers at a time (1-15) by changing the 'D' value.
2.You can Start at a register number other than 0 by changing the 'F' value.
3.To write/read to a FE board register the clock must be on to the backplane,(put 63 8;)

4.There are 7 cores defined in the CCM(6 for FE boards and 1 for the CCM itself) each core can have 3 FE boards.

READING THE REGISTERS
To read front end registers you need to do an array write - the array write tells the CCM which Front End Board - IC - Registers, to read.
The results of the read are stored back into the CCM registers beginning with register number 4.

A: w - write array,
As a note you use array writes(w command) to communicate with FE boards I2C registers.
You have to use array write even if reading only one register.
B: Register 0 of the CCM, must be 0 for an array write..
C: Core number - use the excel file that details the systems core numbers plus add 128.
D: Number of reads that you are doing
E: I2C address(basically selects the board/IC combination that you want to communicate with.
F: Beginnig register number that you want to read.
You follow the array write command with "get" commands, the values are stored in the CCM registers beginning with register 4.

A B C D E F

```
w 0 160 4 0 0; Read core 6(160=32 + 128), board 1 CCA1 register 0-3
get 4; board 1 CCA1 register 0 value
get 5; board 1 CCA1 register 1 value
get 6; board 1 CCA1 register 2 value
get 7; board 1 CCA1 register 3 value
w 0 132 5 18 0; Read core 3(132=4 + 128), board 2 CCA3 register 0-4
get 4; board 2 CCA3 register 0 value
get 5; board 2 CCA3 register 1 value
get 6; board 2 CCA3 register 2 value
get 7; board 2 CCA3 register 3 value
get 8; board 2 CCA3 register 4 value
w 0 129 2 33 1; Read core 1(129=1 + 128), board 3 CCA2 register 1-2
get 4; board 2 CCA2 register 1 value
get 5; board 2 CCA2 register 2 value
```

SH 7-28-2004